Privileged Access Service

Centrify Platform Events and ArcSight CEF Guide

January 2021

Centrify Corporation

Legal Notice

This document and the software described in this document are furnished under and are subject to the terms of a license agreement or a non-disclosure agreement. Except as expressly set forth in such license agreement or non-disclosure agreement, Centrify Corporation provides this document and the software described in this document "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. Some states do not allow disclaimers of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This document and the software described in this document may not be lent, sold, or given away without the prior written permission of Centrify Corporation, except as otherwise permitted by law. Except as expressly set forth in such license agreement or non-disclosure agreement, no part of this document or the software described in this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, or otherwise, without the prior written consent of Centrify Corporation. Some companies, names, and data in this document are used for illustration purposes and may not represent real companies, individuals, or data.

This document could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein. These changes may be incorporated in new editions of this document. Centrify Corporation may make improvements in or changes to the software described in this document at any time.

U.S. Government Restricted Rights: If the software and documentation are being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), in accordance with 48 C.F.R. 227.7202-4 (for Department of Defense (DOD) acquisitions) and 48 C.F.R. 2.101 and 12.212 (for non-DOD acquisitions), the government's rights in the software and documentation, including its rights to use, modify, reproduce, release, perform, display or disclose the software or documentation, will be subject in all respects to the commercial license rights and restrictions provided in the license agreement.

Centrify, DirectControl, DirectAuthorize, DirectAudit, DirectSecure, DirectControl Express, Centrify for Mobile, Centrify for SaaS, DirectManage, Centrify Express, DirectManage Express, Centrify Suite, Centrify User Suite, Centrify Identity Service, Centrify Privilege Service and Centrify Server Suite are registered trademarks of Centrify Corporation in the United States and other countries. Microsoft, Active Directory, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

Centrify software is protected by U.S. Patents 7,591,005; 8,024,360; 8,321,523; 9,015,103; 9,112,846; 9,197,670; 9,442,962 and 9,378,391.

The names of any other companies and products mentioned in this document may be the trademarks or registered trademarks of their respective owners. Unless otherwise noted, all of the names used as examples of companies, organizations, domain names, people and events herein are fictitious. No association with any real company, organization, domain name, person, or event is intended or should be inferred.

Contents

Introduction

The *Centrify Platform Events and ArcSight CEF Guide* is written to provide detailed instructions for accessing events from the Centrify Platform using REST APIs. The guide also presents instructions for creating ArcSight Common Event Format (CEF) Centrify Platform events.

Overview of the steps for accessing Centrify Platform events

The general steps that you perform to access Centrify Platform events are as follows:

1. As a prerequisite to accessing Centrify Platform events, configure the tenant for OAuth access to create:
    - SIEM user
    - OAuth app
    - SIEM scope for accessing Redrock and query
2. Generate the basic authorization token.
3. Fetch the OAuth access token using the oauth2/token API.
4. Fetch the Centrify Platform events using the Redrock/query API.
5. Parse the response that was received from the Redrock/query API.

Prerequisite for accessing Centrify Platform events

The first task that you must perform before accessing Centrify Platform events is to configure the OAuth tenant. For detailed steps, see Setting up the SIEM User and the OAuth App on the Tenant.

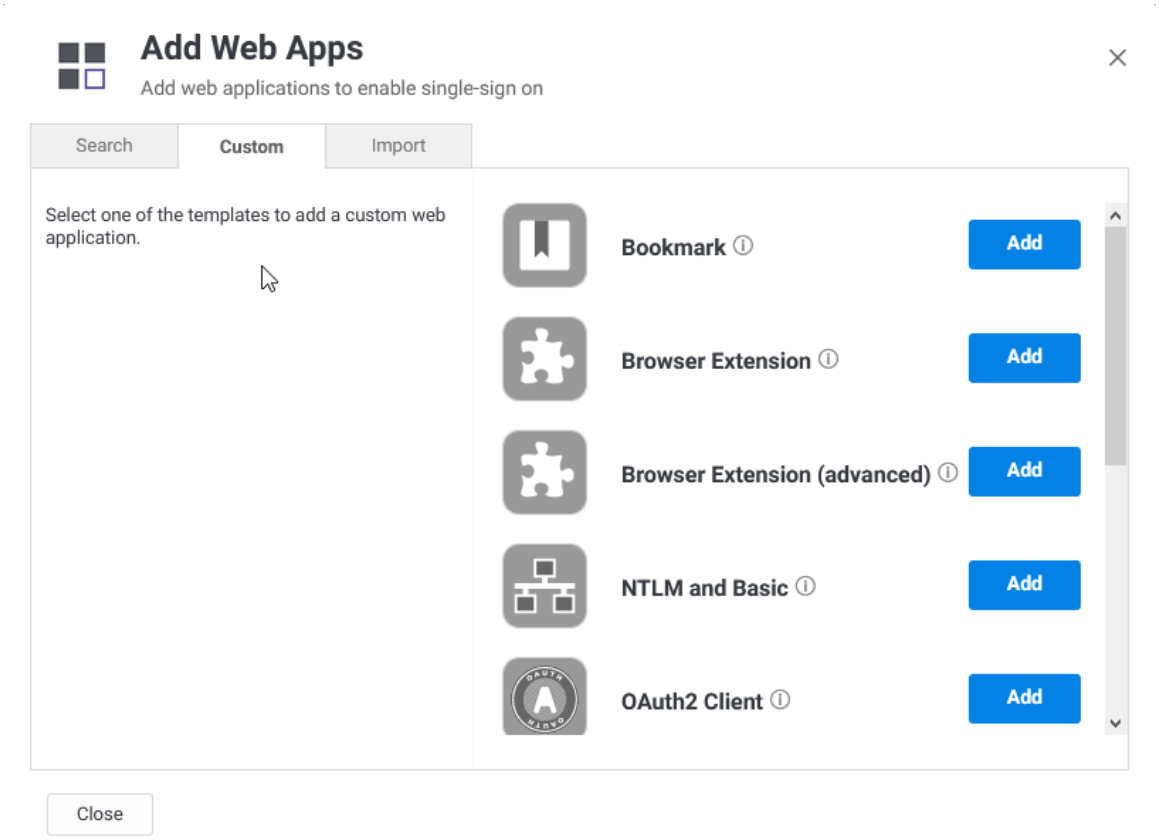
After you complete the configuration, you will have created the following:

- SIEM user
- OAuth app
- SIEM scope for accessing Redrock and query

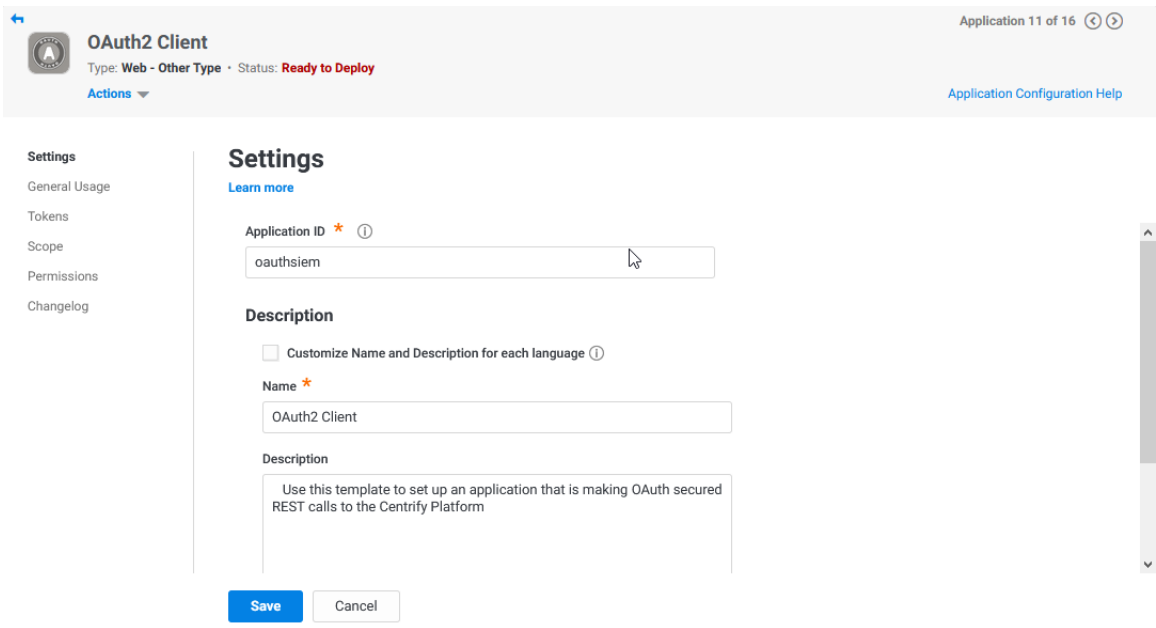
Setting up the SIEM user and the OAuth app on the tenant

To set up the SIEM user and OAuth app

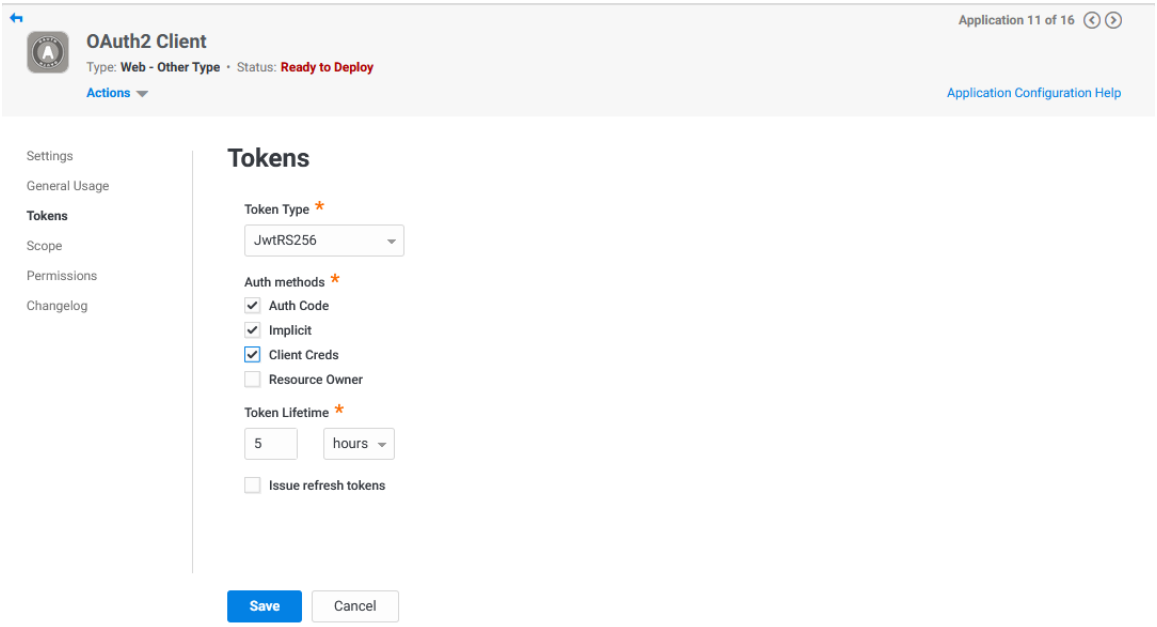
1. In the Admin Portal, open the **Apps** tab and click **Web Apps**.
2. Click **Add Web Apps**.
3. Select the **Custom** tab and click **Add** for OAuth2 Client.



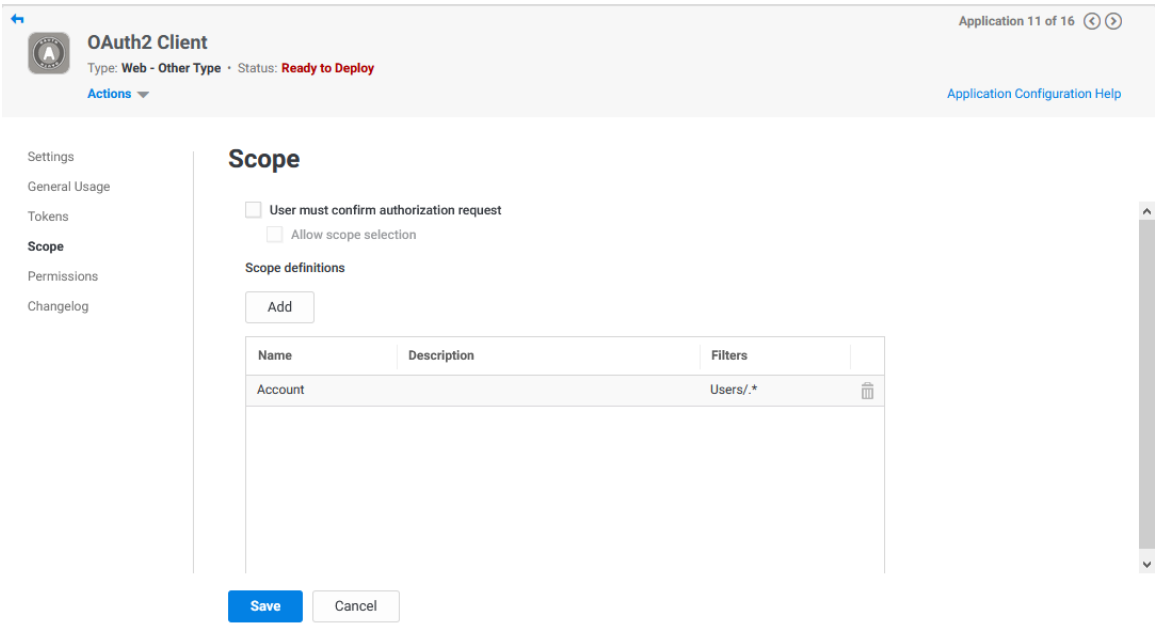
4. When prompted to add the OAuth2 Client web app, click **Yes**.
5. Navigate back to the Web Apps screen and click on **OAuth2 Client** to open the app.
6. On the Settings tab, enter **oauthsiem** for the Application ID.



7. On the General Usage tab, leave the defaults as shown.
8. On the Tokens tab, under Auth methods, check **Client Creds**.

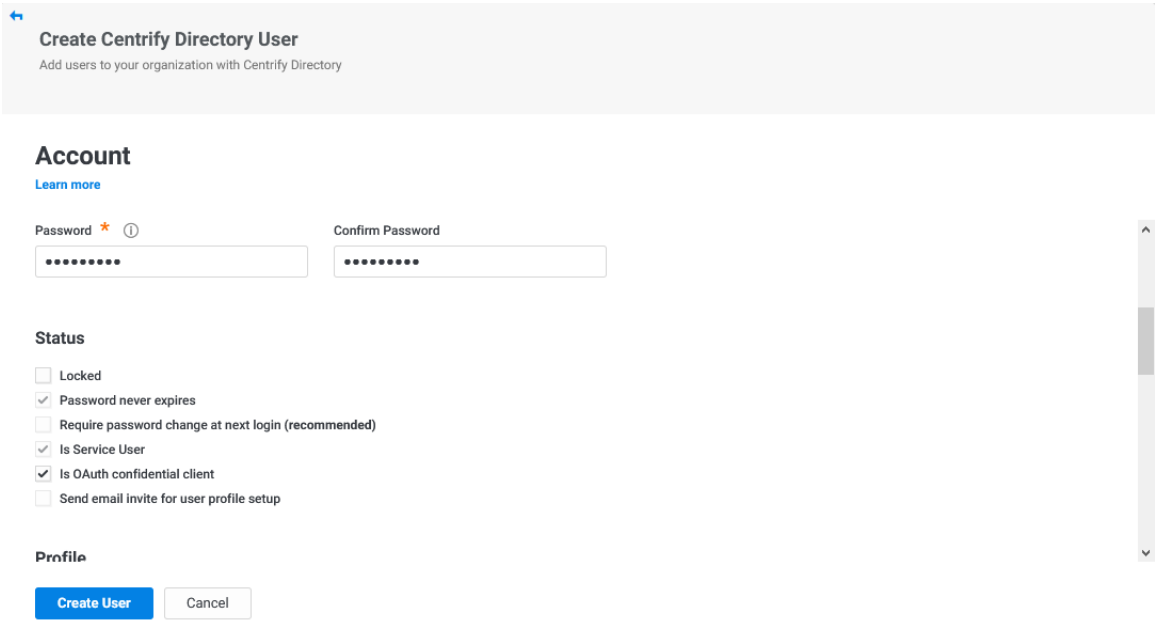


9. On the Scope tab, under Scope definitions, click **Add** to add a new scope.



10. On the Scope definitions dialog:
    a. In the Name field, enter **siem**.
    b. Under Allowed REST APIs, click **Add,** and enter **Redrock/query**.
    c. Click **Save**.

11. Click **Save** to save the OAuth2 Client changes.
12. From the main menu, open **Access** and select the **Users** tab.
13. Click **Add User** to add a new user.
14. On the Create Centrify Directory User page, fill out the following fields:
    a. For the Login Name, enter **siemuser**.
    b. For the Suffix, select **centrify.com** (or leave as is).
    c. For Email Address and Display name, enter the user's email address and full name.
    d. Scroll down to the other account fields:



    e. For the Password and Confirm Password, enter the password of your choice. The password must be between 4 to 64 characters long and contain at least one digit.
    f. Under Status, check **Is OAuth confidential client**. This selection should automatically check**Password never expires**.
    g. Click **Create User** to create the new user.
15. From the main menu, select the **Roles** tab and click **Add Rule**.
16. On the service account page:
    a. On the Description tab enter: **service account** for the Name field. This entry serves as the role name.

b. On the Members tab, search for the **siemuser** that you created earlier and select its checkbox to add the new member.

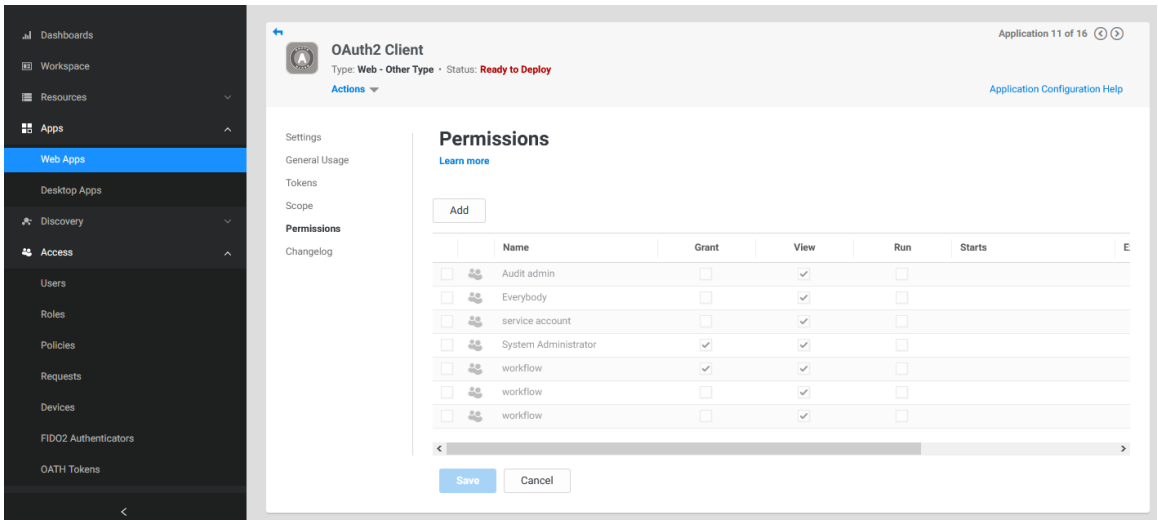
c. Click **Add**.

17. On the Administrative Rights page, click **Add** to open the Add Rights list.
18. Check **Read Only System Administration** and click **Add**.
19. Check **Read Only System Administration** and click **Save**.
20. Perform final checks to ensure:

- On the Users tab, the **siemuser** is shown.

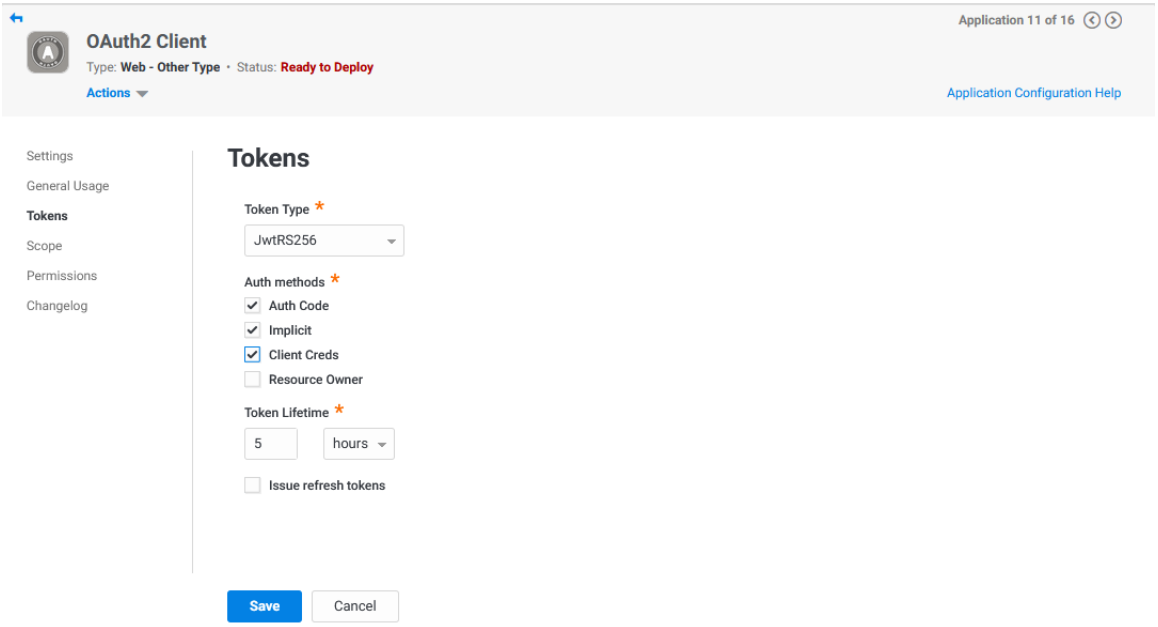
  Note: You may need to check **All Users** to ensure you are shown the full list of users.

- When you select **siemuser** and click on the **Roles** section, **service account**is listed.
- Select **Web Apps** > **OAuth2 Client** > **Permissions** . Ensure the permissions for the**service account** role is shown.



- Select the **Tokens** tab and ensure **Client Creds** is checked under

Auth methods.



Generating a basic authorization token

To generate a basic authorization token, use the following command:

```
echo -ne "<siem_user>:<password>" | base64
```

Example

Review the following example:

```
echo -ne siemuser@centrify.com:Pass@2k17" | base64
```

Sample output

The sample output looks like this:

```
c2llbXVzZXJAY2VudHJpZnkuY29tOkxlZW5hQDIwMTc
```

Fetching the OAuth access token by using the oauth2/token API

Use the curl command and the basic authorization token extracted in the previous step:

```
curl -H "Authorization: Basic <basic_auth_token>" -H "X-CENTRIFY-NATIVE-
CLIENT:True -d "grant_type=client_credentials&scope=<siem_scope>"
https://<tenant>/oauth2/token/<oauth_app_id>
```

Sample curl command

Review the sample curl command:

```
curl -H "Authorization: Basic c2llbXVzZXJAY2VudHJpZnkuY29tOkalZW5hQDIwMTc" -H
"X-CENTRIFY-NATIVE-CLIENT:True -d "grant_type=client_credentials&scope=siem"
https://aaa0056.my-dev.centrify.com/oauth2/token/oauthsiem
```

Sample output

The sample output looks like this:

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6Ijk5QzA4QjQzMjk4N0ZDQjRCN0E5MTEwMTd
DMTI3QzA4NTZCMjAxQzkiLCJ4NXQiOiJtY0NMUXltSF9MUzNxUkVCZkJKOENGYXlBY2siLCJhcHBfaW
QiOiJvYXV0aHNpZW0ifQ.eyJpYXQiOjE1MjE2OTkzNzgsInVuaXF1ZV9uYW1lIjoic2llbXVzZXJAY2
VudHJpZnkuY29tIiwiZXhwIjoxNTIxNzE3Mzc4LCJzdWIiOiI0NDZjOTc5Ni1lOWE4LTRiMDgtYmJkZ
i02ZGZlNTJiOGRkOTIiLCJzY29wZSI6InNpZW0ifQ.e5oE58Cxv0qkIb1Z-nCXyhbIxcL_
6Bs3znVVyBG6aFb6oHSlb_
y5pPnWaLfQdmfnx6hyHtM0GGRoK6HTVJulSbrCFzqHKBHoW38YPh5M7IzTJflJ-
8k0ip9we3ElWm2QiOcbR8AmULYaDR8OnvpIVtmBJ2ZBJng9oFippwoNtBi2gYFjjJsGtRClpqvlHrTy
tPAqe3SvM0whm8yfbq8YhIapcdk_mfJl2YEPX_pyl-Kxzyz9_nHw-_jm0LXzMazvPiAz-
sFCrc8ngtzQZgvDe1wUnPqqEiB0G2Hg2-
NCPYi9hcR8OUyeKD4erkgyXRq1KvvrS7G9iLHT1VrLSu0o2g
```

Fetching events by using the Redrock/query API

Use the curl command and the OAuth access token extracted in the previous step:

```
curl  -H "Authorization: Bearer <oauth_access_token>" -H "X-CENTRIFY-NATIVE-
CLIENT:True -d '{"Script":"<query>"}' https://<tenant>/Redrock/query
```

Sample curl commands

This sample curl command fetches events for the last 24 hours:

```
curl  -H "Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6Ijk5QzA4QjQzMjk4N0ZDQjRCN0E5MTEwMTd
DMTI3QzA4NTZCMjAxQzkiLCJ4NXQiOiJtY0NMUXltSF9MUzNxUkVCZkJKOENGYXlBY2siLCJhcHBfaW
QiOiJvYXV0aHNpZW0ifQ.eyJpYXQiOjE1MjE2OTkzNzgsInVuaXF1ZV9uYW1lIjoic2llbXVzZXJAY2
VudHJpZnkuY29tIiwiZXhwIjoxNTIxNzE3Mzc4LCJzdWIiOiI0NDZjOTc5Ni1lOWE4LTRiMDgtYmJkZ
i02ZGZlNTJiOGRkOTIiLCJzY29wZSI6InNpZW0ifQ.e5oE58Cxv0qkIb1Z-nCXyhbIxcL_
6Bs3znVVyBG6aFb6oHSlb_
y5pPnWaLfQdmfnx6hyHtM0GGRoK6HTVJulSbrCFzqHKBHoW38YPh5M7IzTJflJ-
8k0ip9we3ElWm2QiOcbR8AmULYaDR8OnvpIVtmBJ2ZBJng9oFippwoNtBi2gYFjjJsGtRClpqvlHrTy
tPAqe3SvM0whm8yfbq8YhIapcdk_mfJl2YEPX_pyl-Kxzyz9_nHw-_jm0LXzMazvPiAz-
sFCrc8ngtzQZgvDe1wUnPqqEiB0G2Hg2-
NCPYi9hcR8OUyeKD4erkgyXRq1KvvrS7G9iLHT1VrLSu0o2g" -H "X-CENTRIFY-NATIVE-
CLIENT:True" -d '{"Script":"Select * from Event where WhenOccurred > datefunc
('\''now'\'', '\''-1'\'')"}' https://aaa0056.my-dev.centrify.com/Redrock/query
```

This sample curl command fetches events between two timestamps:

```
curl  -H "Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6Ijk5QzA4QjQzMjk4N0ZDQjRCN0E5MTEwMTd
DMTI3QzA4NTZCMjAxQzkiLCJ4NXQiOiJtY0NMUXltSF9MUzNxUkVCZkJKOENGYXlBY2siLCJhcHBfaW
QiOiJvYXV0aHNpZW0ifQ.eyJpYXQiOjE1MjE2OTkzNzgsInVuaXF1ZV9uYW1lIjoic2llbXVzZXJAY2
VudHJpZnkuY29tIiwiZXhwIjoxNTIxNzE3Mzc4LCJzdWIiOiI0NDZjOTc5Ni1lOWE4LTRiMDgtYmJkZ
i02ZGZlNTJiOGRkOTIiLCJzY29wZSI6InNpZW0ifQ.e5oE58Cxv0qkIb1Z-nCXyhbIxcL_
6Bs3znVVyBG6aFb6oHSlb_
y5pPnWaLfQdmfnx6hyHtM0GGRoK6HTVJulSbrCFzqHKBHoW38YPh5M7IzTJflJ-
8k0ip9we3ElWm2QiOcbR8AmULYaDR8OnvpIVtmBJ2ZBJng9oFippwoNtBi2gYFjjJsGtRClpqvlHrTy
tPAqe3SvM0whm8yfbq8YhIapcdk_mfJl2YEPX_pyl-Kxzyz9_nHw-_jm0LXzMazvPiAz-
sFCrc8ngtzQZgvDe1wUnPqqEiB0G2Hg2-
NCPYi9hcR8OUyeKD4erkgyXRq1KvvrS7G9iLHT1VrLSu0o2g" -H "X-CENTRIFY-NATIVE-
CLIENT:True -d '{"Script":"Select * from Event where WhenOccurred >= '\''2018-
03-15T11:33:59.273000Z'\'' and  WhenOccurred < '\''2018-03-
21T11:33:59.273000Z'\''"}' https://aaa0056.my-dev.centrify.com/Redrock/query
```

Parsing the response received from Redrock/query

Refer to the following sample Python code to extract events data from a response:

```
import json
response_json = json.loads(response.text)
events = response_json['Result']['Results']
headers = []
for column in response_json['Result']['Columns']:
    headers.append(column['Name'])
for idx, event in enumerate(events):
    print('\n Row Number:' + str(idx))
    for header in headers:
        if event['Row'][header] is not None:
            print(header + "=" + str(event['Row'][header]))
```

References

For additional information, see:

https://developer.centrify.com/docs/use-queries

https://docs.centrify.com/Content/CoreServices/Reports/FilterEvents.htm

ArcSight CEF format

The Common Event Format (CEF) standard format, developed by ArcSight, enables vendors and their customers to quickly integrate their product information into ArcSight ESM.

CEF defines a syntax for log records comprised of a standard header and a variable extension, formatted as key-value pairs.

When syslog is used as a transport mechanism, CEF uses the following format, comprised of a syslog prefix, a header, and an extension:

```
Jan 18 11:07:53 host CEF:Version|Device Vendor|Device
Product|Device
Version|Device Event Class
ID|Name|Severity|[Extension]
```

The following example illustrates a general CEF message using syslog transport:

```
Sep 19 08:26:10 host
CEF:0|Centrify|Centrify_
Cloud|1.0|Cloud.core|Cloud.core.MfaSummary|5|src=10.0.0.1
dst=2.1.2.2 spt=1232
```

Using CEF without wyslog

Syslog applies a syslog prefix to each message, no matter what device it arrives from, which contains the date and hostname:

```
Jan 18 11:07:53 host CEF:Version|...
```

However, if an event producer is unable to write syslog messages, it is still possible to write the events to a file. In this case, begin the message with the format shown below, and omit the syslog prefix:

```
CEF:Version|Device Vendor|Device Product|Device Version|Device
Event Class ID|Name|Severity|[Extension]
```

Sample Python functions for CEF creation

This section describes a set of sample Python functions for generating CEF-formatted CP events.

There are three main functions in this package:

- `fetch_oauth_token()`
- `query_events()`
- `cef_generator()`

Using the functions to demonstrate sample usage

Prerequisite: Python 3.5 or above

Follow these steps:

1. Download the Python code from https://github.com/centrify/centrify-hparcsight-integration-sample/
2. Install pip packages in `requirement.txt`.
3. Provide the values for `tenant`, `siem_username`, and `siem_password` in `config.ini`.
4. Execute `sample_usage.py` to generate CEF-formatted CP events for one hour:

   ```
   python3.5 sample_usage.py
   ```

The following example shows a CEF message for a Self-Service App Launch CIS Event:

```
CEF:0|Centrify|Centrify_
Cloud|1.0|Cloud.Saas.Application|Cloud.Saas.Application.SelfServiceAppLaunch|5|
dhost=AAA0056 duser=cloudadmin@persistent.com01 msg=User
cloudadmin@persistent.com01 launched Instagram from 103.6.32.100
shost=103.6.32.100 src=103.6.32.100 rt=1525844566655 deviceProcessName=centrify-
syslog-writer dvchost=dinesh-VirtualBox dtz=Africa/Abidjan
requestContext=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/52.0.2743.116 Safari/537.36 Edge/15.15063
externalId=772a4a904e82da87.W00.0315.1aa20afe647f09c dpriv=WebRole
destinationServiceName=CDS suid=c2c7bcc6-9560-44e0-8dff-5be221cd37ee
cs1=Instagram cs1Label=applicationId cs2=Instagram cs2Label=applicationName
cs3=Web cs3Label=applicationType cs4=103.6.32.100 cs4Label=clientIPAddress
cs5=65f79bb1-4f91-4496-9991-d148da16cc3e cs5Label=internalSessionId
cs6=0d10a24f4c57434198fb3ad4559cc48b cs6Label=azDeploymentId
directoryServiceNameLocalized=Centrify Directory threadType=RestCall
azRoleId=WebRole_IN_0 internalTrackingID=d3a0713b610146ca916155efca2be690
authMethod=UserPassword requestIsMobileDevice=False
directoryServiceUuid=09B9A9B0-6CE8-465F-AB03-65766D33B05E
requestDeviceOS=Windows level=Info
```

You can customize the usage or the APIs per your application needs.

Note: CEF has a predefined set of keys.

CEF mapping of CP events

This section provides detailed information about how the CEF fields have been mapped from the CP event fields in the Python application described above.

CEF header

Header Field	CP Event Field
Version	‘0’
Device Vendor	‘Centrify’
Device Product	‘Centrify_Cloud’
Device Version	‘1.0’
Device Event Class ID	**Variable — depends on the event.** For example: ‘Cloud.Saas.Application’

Header Field	CP Event Field
Name	**Variable — depends on the event.** For example: Cloud.Saas.Application.SelfServiceAppLaunch’
Severity	**Variable — depends on the Level field in event.** For example: ‘5’ for Info, ‘10’ for Error.

CP ArcSight CEF extension

The CEF Extension contains a collection of key-value pairs. The keys are predefined and are referred to as the *ArcSight Extension Dictionary*. (CEF Fields)

Common properties in CP events

This section lists the CEF field mapping of CP events, which are part of the CEF extension.

These properties are common to all events of the Centrify Platform and Privilege Services.

ArcSight CEF Field	CP Event Field
The common properties are those listed below in **bold**.	
Destination Host Name	Tenant
Destination User Name	NormalizedUser
Message	EventMessage
Source Host Name	RequestHostName
Source Address	FromIPAddress

ArcSight CEF Field	CP Event Field
Device Receipt Time	whenoccurred_epoch_ms (This is the event timestamp in UTC)
Device Process Name	'centrify-syslog-writer' (can be configured in cef_mapping.ini)
Device Host Name	Hostname of machine running the python app
Device Time Zone	'Africa/Abidjan' **Note:** (This time zone is chosen mainly to set UTC offset to 0)
The keys in the common properties section below are added in the CEF message **only if** no event-specific CEF mapping is specified for an event in the mapping configuration file, which is enclosed with the Sample Python application for CEF creation.	
Device Custom String 1	AuthMethod
Device Custom String1 Label	'authMethod'
Device Custom String2	RequestIsMobileDevice
Device Custom String2 Label	'requestIsMobileDevice'
Device Custom String3	DirectoryServiceUuid
Device Custom String3 Label	'directoryServiceUuid'
Device Custom String4	RequestDeviceOS
Device Custom String4 Label	'requestDeviceOS'
Device Custom String5	Level

ArcSight CEF Field	CP Event Field
Device Custom String5 Label	Level

Event-specific properties in CP

This section lists the event-specific properties mapped to ArcSight Fields. All events (whether they are listed below or not) will have the first nine common properties, identified in the table above, mapped in an ArcSight.CEF message. The message is generated when you use the Sample Python functions described earlier in this document.

Any CEF key appearing in event-specific mapping will override the CEF key mapping in the common properties section. For example, the Cloud.Server.ManualAccount.SessionStart event, Destination host (Dhost), and Destination User(duser) will be `'ComputerName'` and `'AccountName'`, which will overwrite the common properties mapped for dhost and duser.

EventType=Cloud.Core.MfaSummary

ArcSight CEF Field	CP Event Field
Reason	MfaReason
Outcome	MfaResult
RequestContext	RequestUserAgent
ExternalId	ID
Dpriv	AzRoleName
DestinationServiceName	DirectoryServiceName
Device Custom String 1	MfaInitiator
Device Custom String1 Label	'mfaInitiator'

ArcSight CEF Field	CP Event Field
Device Custom String2	FactorsLocalized
Device Custom String2 Label	‘factorsLocalized’
Device Custom String3	ProfileName
Device Custom String3 Label	‘profileName’
Device Custom String4	FailReason
Device Custom String4 Label	‘failReason’
Device Custom String5	MfaUnlock
Device Custom String5 Label	‘mfaUnlock’
Device Custom String6	ForgotPassword
Device Custom String6 Label	‘forgotPassword’
Device Custom Number1	Factorcount
Device Custom Number1 Label	‘factorCount’
Device Custom Number2	SecurityQuestionAnswerCount
Device Custom Number2 Label	‘securityQuestionAnswercount’

Note: The remaining fields in an event that are not mapped to CEF keys will still be added in the CEF message with their CP-event field keys. These custom non-CEF keys will not be available for reporting in ArcSight, but they can viewed as part of the raw event message.

EventType=Cloud.Saas.Application.AppLaunch

ArcSight CEF Field	CP Event Field
RequestContext	RequestUserAgent
ExternalId	ID
Dpriv	AzRoleName
DestinationServiceName	DirectoryServiceName
Suid	UserGuid
Device Custom String 1	ApplicationID
Device Custom String1 Label	‘applicationId’
Device Custom String2	ApplicationName
Device Custom String2 Label	‘applicationName’
Device Custom String3	ApplicationType
Device Custom String3 Label	‘applicationType’
Device Custom String4	TemplateName
Device Custom String4 Label	‘templateName’
Device Custom String5	InternalSessionId
Device Custom String5 Label	‘internalSessionId’
Device Custom String6	AzDeploymentId
Device Custom String6 Label	azDeploymentId

EventType=Cloud.Saas.Application.GatewayAppLaunch

EventType=Cloud.Saas.Application.SelfServiceAppLaunch

ArcSight CEF Field	CP Event Field
RequestContext	RequestUserAgent
ExternalId	ID
Dpriv	AzRoleName
DestinationServiceName	DirectoryServiceName
Suid	UserGuid
Device Custom String 1	ApplicationID
Device Custom String1 Label	‘applicationId’
Device Custom String2	ApplicationName
Device Custom String2 Label	‘applicationName’
Device Custom String3	ApplicationType
Device Custom String3 Label	‘applicationType’
Device Custom String4	ClientIPAddress
Device Custom String4 Label	‘clientIPAddress’
Device Custom String5	InternalSessionId
Device Custom String5 Label	‘internalSessionId’
Device Custom String6	AzDeploymentId

ArcSight CEF Field	CP Event Field
Device Custom String6 Label	azDeploymentId

EventType=Cloud.Server.ManualAccount.SessionStart

EventType= Cloud.Server.LocalAccount.SessionStart

ArcSight CEF Field	CP Event Field
Src	FromIPAddress
Suser	NormalizedUser
Dhost	ComputerName
Duser	AccountName
RequestContext	RequestUserAgent
ExternalId	ID
Dpriv	AzRoleName
DestinationServiceName	DirectoryServiceName
Suid	UserGuid
Device Custom String 1	UserType
Device Custom String1 Label	‘userType’

ArcSight CEF Field	CP Event Field
Device Custom String2	SessionType
Device Custom String2 Label	‘sessionType’
Device Custom String3	AuthorityName
Device Custom String3 Label	‘authorityName’
Device Custom String4	JumpType
Device Custom String4 Label	‘jumpType’
Device Custom String5	DirectoryServiceNameLocalized
Device Custom String5 Label	‘directoryServiceNameLocalized’
Device Custom String6	AuthoritySource
Device Custom String6 Label	‘authoritySource’

EventType=Cloud.Server.LocalAccount.PasswordExport

EventType= Cloud.Server.DomainAccount.PasswordExport

ArcSight CEF Field	CP Event Field
Src	FromIPAddress
Suser	NormalizedUser

ArcSight CEF Field	CP Event Field
Dhost	ComputerName
Duser	AccountName
RequestContext	RequestUserAgent
ExternalId	ID
Dpriv	AzRoleName
DestinationServiceName	DirectoryServiceName
Suid	UserGuid
Device Custom String 1	UserType
Device Custom String1 Label	‘userType’
Device Custom String2	AuthorityID
Device Custom String2 Label	‘authorityID’
Device Custom String3	AuthorityName
Device Custom String3 Label	‘authorityName’
Device Custom String4	AzRoleId
Device Custom String4 Label	‘azRoleId’
Device Custom String5	DirectoryServiceNameLocalized
Device Custom String5 Label	‘directoryServiceNameLocalized’
Device Custom String6	CheckedOut
Device Custom String6 Label	‘checkedOut’

ArcSight CEF Field	CP Event Field
Device Custom Date1	WhenDueBack
Device Custom Date1 Label	‘whenDueBack’

EventType=Cloud.Core.Server.CpsTileLaunch

ArcSight CEF Field	CP Event Field
RequestContext	RequestUserAgent
ExternalId	ID
Dpriv	AzRoleName
DestinationServiceName	DirectoryServiceName
Suid	UserGuid
Device Custom String 1	UserType
Device Custom String1 Label	‘userType’
Device Custom String2	ApplicationType
Device Custom String2Label	‘applicationType’
Device Custom String3	ApplicationName
Device Custom String3Label	‘applicationName’
Device Custom String4	ApplicationID
Device Custom String4Label	‘applicationId’

ArcSight CEF Field	CP Event Field
Device Custom String5	DirectoryServiceNameLocalized
Device Custom String5Label	‘directoryServiceNameLocalized’
Device Custom String6	InternalTrackingID
Device Custom String6Label	‘internalTrackingID’

EventType=Cloud.Core.AdaptiveMfa.RiskAnalysis

Only Common properties.

Alternate approach for creating the Common Extension Format (CEF)

In case you are using the CP REST APIs directly in your application and generating your own Centrify Platformsyslog messages in a generic non-CEF format having key=value pairs separated by a delimiter, then ArcSight SmartConnector will need to be installed and configured to collect these Centrify Platform syslog.

These logs will need to be parsed into CEF format by creating ArcSight FlexConnector, to enable the Centrify Platform events to be usable for SIEM in ArcSight. The only downside to using a FlexConnector is that ArcSight does not officially certify it.